

### **Amendments to the Claims:**

This listing of claims will replace all prior versions, and listings, of claims in the application:

### **Listing of Claims:**

1. (Currently Amended) A computer implemented method[[.]] ~~in a data processing system~~, for identifying differences between [[the]] an execution of a first ~~build of a built~~ computer program and a second ~~build of a built~~ computer program, comprising:

obtaining a first call tree data structure corresponding to first trace data of [[an]] an execution of the first ~~build of a built~~ computer program;

obtaining a second call tree data structure corresponding to second trace data of [[an]] another execution of the second ~~build of a built~~ computer program;

~~copying the first call tree data structure to form a copied call tree data structure;~~

walking subtracting the second call tree data structure from over the copied call first call tree data structure to generate a subtracted third call tree data structure, wherein the third call tree data structure includes all nodes of both the first call tree data structure and the second call tree data structure, and wherein each node of the third call tree data structure includes a pass field having one of a first pass field value indicating that a first node was only present in the first call tree data structure, a second pass field value indicating that a second node was only present in the second call tree data structure, and a third pass field value indicating that a third node was present in both the first call tree data structure and the second call tree data structure; and

outputting the ~~subtracted third~~ third call tree data structure, wherein the ~~subtracted third~~ third call tree data structure identifies differences between the execution of the first ~~build of a built~~ computer program and the another execution of the second ~~build of a built~~ computer program.

2. (Original) The method of claim 1, further comprising:

inputting the first trace data and the second trace data to an arcflow tool, wherein the arcflow tool generates the first call tree data structure and the second call tree data structure based on the first trace data and the second trace data.

3. (Original) The method of claim 1, wherein the first call tree data structure and the second call tree data structure are xtree data structures.

4. (Currently Amended) The method of claim 1, ~~wherein copying the first call tree data structure to form the copied call tree data structure includes~~ further comprising:

inserting ~~[[a]]~~ the pass field in each node of the ~~copied first~~ first call tree data structure; and  
initializing the pass field in each node of the ~~copied first~~ first call tree data structure to have the first pass field value[[;]] and  
~~walking the second call tree data structure over the copied call tree data structure to generate the subtracted call tree data structure.~~

5. (Currently Amended) The method of claim ~~[[4]]~~ 1, wherein walking the second call tree data structure over the ~~copied first~~ first call tree data structure includes:

for each node that exists in both the ~~copied first~~ first call tree data structure and the second call tree data structure, generating a node in the ~~subtracted third~~ third call tree data structure by subtracting a base value of the node in the second call tree data structure from a base value of a corresponding node in the ~~copied first~~ first call tree data structure.

6. (Currently Amended) The method of claim ~~[[4]]~~ 1, wherein walking the second call tree data structure over the first call tree data structure includes:

for each node that exists in only the second call tree data structure, creating a node in the ~~subtracted third~~ third call tree data structure having a negative base value corresponding to a base value of the node that exists in only the second call tree data structure.

7. (Currently Amended) The method of claim 5, further comprising:

setting ~~[[a]]~~ the value of a pass field of the node in the ~~subtracted third~~ third call tree data structure to the third pass field value ~~a value indicating that both the first call tree data structure and the second call tree data structure contributed to the base value.~~

8. (Currently Amended) The method of claim 6, further comprising:

setting ~~[[a]]~~ the value of a pass field of the node in the ~~subtracted third~~ third call tree data structure to ~~[[a]]~~ the second pass field value ~~indicating that either the first call tree data structure or the second call tree data structure contributed to the base value.~~

9. (Currently Amended) The method of claim 4, wherein nodes in the first call tree data structure and nodes in the second call tree data structure whose paths have not changed between builds are not present in the ~~subtracted third~~ third call tree data structure.

10. (Currently Amended) The method of claim 1, further comprising:

identifying improvements or regressions from the first ~~build~~ built computer program to the second ~~build of a~~ built computer program based on values associated with nodes in the ~~subtracted~~ third call tree data structure.

11. (Currently Amended) A computer program product in a computer readable medium for identifying differences between the execution of a first ~~build of a~~ built computer program and a second ~~build of a~~ built computer program, comprising:

first instructions for obtaining a first call tree data structure corresponding to first trace data of an execution of the first ~~build of a~~ built computer program;

second instructions for obtaining a second call tree data structure corresponding to second trace data of ~~[[an]]~~ another execution of the second ~~build of a~~ built computer program;

~~fifth instructions for copying the first call tree data structure to form a copied call tree data structure;~~

third instructions for walking ~~subtracting~~ the second call tree data structure ~~[[from]]~~ over the ~~copied call~~ first call tree data structure to generate a ~~subtracted~~ third call tree data structure, wherein the third call tree data structure includes all nodes of both the first call tree data structure and the second call tree data structure, and wherein each node of the third call tree data structure includes a pass field having one of a first pass field value indicating that a first node was only present in the first call tree data structure, a second pass field value indicating that a second node was only present in the second call tree data structure, and a third pass field value indicating that a third node was present in both the first call tree data structure and the second call tree data structure; and

fourth instructions for outputting the ~~subtracted~~ third call tree data structure, wherein the ~~subtracted~~ third call tree data structure identifies differences between the execution of the first ~~build of a~~ built computer program and the another execution of the second ~~build of a~~ built computer program.

12. (Original) The computer program product of claim 11, further comprising:

fifth instructions for inputting the first trace data and the second trace data to an arcflow tool, wherein the arcflow tool generates the first call tree data structure and the second call tree data structure based on the first trace data and the second trace data.

13. (Original) The computer program product of claim 11, wherein the first call tree data structure and the second call tree data structure are xtree data structures.

14. (Currently Amended) The computer program product of claim 11, ~~wherein the fifth instructions for copying the first call tree data structure to form the copied call tree data structure include further~~ comprising:

sixth instructions for inserting ~~[[a]]~~ the pass field in each node of the ~~copied~~ first call tree data structure; and

seventh instructions for initializing the pass field in each node of the ~~copied~~ first call tree data structure to have the first pass field value~~[[;]]~~ and

~~instructions for walking the second call tree data structure over the copied call tree data structure to generate the subtracted call tree data structure.~~

15. (Currently Amended) The computer program product of claim ~~[[14]]~~ 11, wherein the third instructions for walking the second call tree data structure over the ~~copied~~ first call tree data structure include:

for each node that exists in both the ~~copied~~ first call tree data structure and the second call tree data structure, instructions for generating a node in the ~~subtracted~~ third call tree data structure by subtracting a base value of the node in the second call tree data structure from a base value of a corresponding node in the ~~copied~~ first call tree data structure.

16. (Currently Amended) The computer program product of claim ~~[[14]]~~ 11, wherein the third instructions for walking the second call tree data structure over the first call tree data structure include:

for each node that exists in only the second call tree data structure, instructions for creating a node in the ~~subtracted~~ third call tree data structure having a negative base value corresponding to a base value of the node that exists in only the second call tree data structure.

17. (Currently Amended) The computer program product of claim 15, further comprising:

instructions for setting ~~[[a]]~~ the ~~value of a~~ pass field of the node in the ~~subtracted~~ third call tree data structure to the third pass field value ~~a value indicating that both the first call tree data structure and the second call tree data structure contributed to the base value.~~

18. (Currently Amended) The computer program product of claim 16, further comprising:

instructions for setting ~~[[a]]~~ the ~~value of a~~ pass field of the node in the ~~subtracted~~ third call tree data structure to ~~[[a]]~~ the second pass field ~~value indicating that either the first call tree data structure or the second call tree data structure contributed to the base value.~~

19. (Currently Amended) The computer program product of claim 11, further comprising:  
~~fifth~~ eighth instructions for identifying improvements or regressions from the first ~~build~~ built computer program to the second ~~build-of-a~~ built computer program based on values associated with nodes in the ~~subtracted~~ third call tree data structure.

20. (Currently Amended) An apparatus for identifying differences between the execution of a first ~~build-of-a~~ built computer program and a second ~~build-of-a~~ built computer program, comprising:  
means for obtaining a first call tree data structure corresponding to first trace data of an execution of the first build of the computer program;  
means for obtaining a second call tree data structure corresponding to second trace data of ~~[[an]]~~ another execution of the second build of the computer program;  
~~means for copying the first call tree data structure to form a copied call tree data structure;~~  
means for ~~subtracting~~ walking the second call tree data structure ~~[[from]]~~ over the ~~copied call first~~ call tree data structure to generate a ~~subtracted~~ third call tree data structure, wherein the third call tree data structure includes all nodes of both the first call tree data structure and the second call tree data structure, and wherein each node of the third call tree data structure includes a pass field having one of a first pass field value indicating that a first node was only present in the first call tree data structure, a second pass field value indicating that a second node was only present in the second call tree data structure, and a third pass field value indicating that a third node was present in both the first call tree data structure and the second call tree data structure; and  
means for outputting the ~~subtracted~~ third call tree data structure, wherein the ~~subtracted~~ third call tree data structure identifies differences between the execution of the first ~~build-of-a~~ built computer program and the another execution of the second ~~build-of-a~~ built computer program.